

PlayStation®2 Development Environment Setup Guide

© 2001 Sony Computer Entertainment Inc.

Publication date: July 2001

Sony Computer Entertainment Inc.
1-1, Akasaka 7-chome, Minato-ku
Tokyo 107-0052, Japan

Sony Computer Entertainment America
919 E. Hillsdale Blvd.
Foster City, CA 94404, U.S.A.

Sony Computer Entertainment Europe
30 Golden Square
London W1F 9LD, U.K.

The *PlayStation®2 Development Environment Setup Guide* manual is supplied pursuant to and subject to the terms of the Sony Computer Entertainment PlayStation® license agreements.

The *PlayStation®2 Development Environment Setup Guide* manual is intended for distribution to and use by only Sony Computer Entertainment licensed Developers and Publishers in accordance with the PlayStation® license agreements.

Unauthorized reproduction, distribution, lending, rental or disclosure to any third party, in whole or in part, of this book is expressly prohibited by law and by the terms of the Sony Computer Entertainment PlayStation® license agreements.

Ownership of the physical property of the book is retained by and reserved by Sony Computer Entertainment. Alteration to or deletion, in whole or in part, of the book, its presentation, or its contents is prohibited.

The information in the *PlayStation®2 Development Environment Setup Guide* manual is subject to change without notice. The content of this book is Confidential Information of Sony Computer Entertainment.

 and PlayStation are registered trademarks of Sony Computer Entertainment Inc. All other trademarks are property of their respective owners and/or their licensors.

Table of Contents

Overview	5
Hardware Setup	6
Development System Architecture Overview	6
Features of the DTL-T10000	6
Equipment Required To Use the Programmer Tools	7
Setting up the DTL-T10000	7
DTL-T10000 Setup Notes	7
Software Setup	9
Development Software Overview	9
Installing Software	9
Installing the compiler	10
Copying IOP Libraries and Header Files	10
Updating dsnet	10
Environment Settings	11
Writing to Flash ROM	11
Confirming the Flash ROM Version	11
Compiling and Running Sample Programs	12
Changing Access Permission Attributes	12
basic3d: Running an EE Sample Program	12
sqhard: Running an IOP Sample Program	13
sqsoft: Running an EE-IOP Sample Program	13

Overview

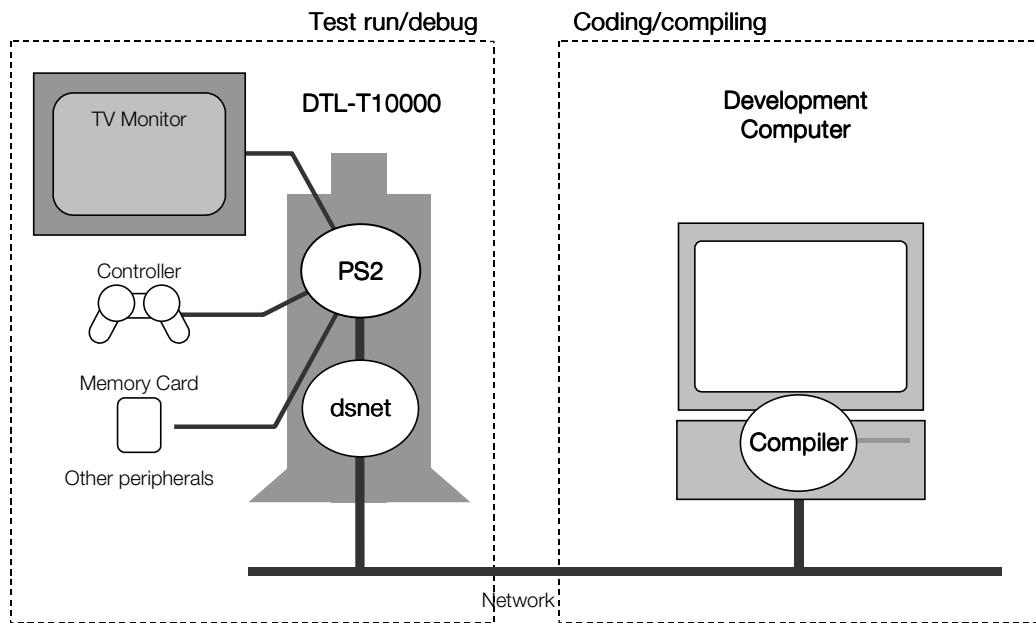
This manual describes the steps required to set up the PlayStation 2 software development environment to compile sample programs. By performing the series of operations described here, the user can confirm that the development environment has been properly set up.

Hardware Setup

Development System Architecture Overview

The following is a schematic drawing showing the development system architecture.

Figure 1: Architecture of the development system



The development system is based on the DTL-T10000/DTL-T10000H PlayStation 2 software development unit (these are both referred to as the DTL-T10000, below). The DTL-T10000 PlayStation 2 software development unit is what is generally referred to in cross-platform development as a target box. The DTL-T10000 is equipped with features corresponding to those of the actual PlayStation 2 as well as administrative and communication functions (dsnet) to allow programs to be test run and debugged. The platform that supports these functions is a single board PC/AT-compatible that runs Linux and is installed in the DTL-T10000.

The DTL-T10000 is a unit designed specifically for test running PlayStation 2 software; the user does not directly log in to the DTL-T10000 to code programs. Coding and compiling are performed on a separate computer (known as a development computer) that is connected to the network.

Features of the DTL-T10000

The DTL-T10000 is a development tool for developing PlayStation 2 software and has the following features:

- It is equipped with the features of the PlayStation 2 and allows efficient software development.
- It comes with a communications processor that allows it to send and receive software and data with a development computer over a network. This allows programs to be run, checked, and debugged efficiently.
- It can be connected directly to a network, so installation and upgrading is simple. This also reduces the load on the development computer.

Equipment Required To Use the Programmer Tools

Hardware (DTL-T10000)

The DTL-T10000 contains a single-board PC (MMX Pentium 233 MHz) serving as the communications processor, but a separate development PC must be available to provide a compiling/debugging environment. A development environment where the DTL-T10000 is remotely accessed through a network is recommended.

Development computer

Any standard personal computer can be used for the compile/debug environment. The following list of specifications should serve as a guideline. The recommended environment is RedHat Linux 5.2. If another version of Linux is to be used, make sure that it uses Libc6.

Table 1: Development computer specifications

Item	Specification
CPU	Pentium II 400 MHz or higher
Memory	128MBytes min.
PC/AT monitor	17 inches min.
Graphics	SVGA

Setting up the DTL-T10000

For additional information on setting up and configuring the DTL-T10000, please refer to the instruction manual that comes with the DTL-T10000. The main procedure is as follows:

1. Connect the TV monitor
2. Connect the unit to the network (see notes below)
3. Turn on the main power supply switch on the back, start by pushing the button on top.
4. Run a web browser on the development computer to access the DTL-T10000
(An administration tool window will be displayed)
5. Use the administration tool screen to set the internal clock, enter network settings, etc.
6. Install "Packages" using the administration tool screen

DTL-T10000 Setup Notes

Connecting to the Network

The setup procedure is a little different depending on whether there is a DHCP server on the network.

If a DHCP server is on the network

If there is a DHCP server on the network, the IP address and subnet mask will be configured automatically when the DTL-T10000 is connected to the network and turned on. However, this will result in the DTL-T10000 having a different IP address each time, so we recommend using a fixed IP address to allow efficient development. To use an automatically assigned IP address permanently (fixed IP address), consult the administrator of the DHCP server.

If a DHCP server is not on the network

If there is no DHCP server on the network (i.e., each device is assigned a permanent IP address), then a local network should be set up and the default IP address should be changed to an IP address that works in the user's environment. The development computer is set up with an IP address of 192.168.0.2 and a subnet mask of 255.255.255.0. The development computer and the DTL-T10000 can then be connected to each other directly using a crossover cable or through straight cables going through a hub that is not already connected to an existing network. The local network will be set up when the development computer and the DTL-T10000 are turned on.

Installing "Packages"

Of the DTL-T10000 internal software, dsnet and the flash ROM are updated as necessary. However, the dsnet tools are installed and updated using the administration tools screen. The most recent version will be distributed along with the development software, so updates should be performed using this. The specific procedures will be described later.

Software Setup

Development Software Overview

SCE provides development software in the form of compilers, libraries, sample programs, and other tools. The main ones are listed below.

- C/C++ compiler
ee-gcc and iop-gcc. Based on gcc and customized for the EE and the IOP.
- Libraries
- Sample programs
- Various tools

Installing Software

The latest versions of development software are distributed on the SCEA and SCEE developer support websites.

The procedure for installing downloaded software is as follows:

Unpack and Install Packages

1. Development software is installed under /usr/local/sce.

When updating, it is safer to save the previous environment by renaming the directory and installing from scratch.

2. As root, unpack in /usr/local.

```
$ su
:
# cd /usr/local
# mv sce sce_old
# tar xzvf lib_110.tgz
:
# tar xzvf eb_110.tgz
:
# ls -1 -t
sce
readme_j.txt
readme.txt
t10000-XXXX.bin
:
:
```

Notes on Installing Software

Development software assumes it is installed in /usr/local/sce, so when installing always unpack using /usr/local as the current directory. If installing elsewhere, create a symbolic link as user root from /usr/local to the other location (e.g., /home/xxx/sce).

```
# ln -s /home/xxx/sce /usr/local
```

Installing the compiler

Download the following compiler from the SCEA or SCEE developer support website.

EE 2.9-ee-991111-01
IOP 2.8.1

Segmented Downloads

Since the size of a package is large, it can be broken into smaller files and downloaded. After downloading the entire module, use the cat command to concatenate each of the files.

Example:

```
$ cat ee9911.0* > ee9911.tgz
```

Uncompress / Install

Gain root privilege. If a previous version already exists, we recommend saving it before performing the installation. The version can also be confirmed using "ee-gcc -v" "iop-gcc -v".

Example:

```
# mv ee9911.tgz /usr/local/
# cd /usr/local/sce/ee
# mv gcc gcc.old (Renamed previous gcc)
# cd /usr/local
# tar zxvf ee9911.tgz
:
$ ee-gcc -v
:
gcc version 2.9-ee-991111-01
```

Copying IOP Libraries and Header Files

Execute the install script sce/iop/install/Install.sh for the headers and static libraries.

Updating dsnet

dsnet on the DTL-T10000 can be updated. Go to "T10000 administration tools" using a web browser on the development machine and specify the following rpm package for updating. Specific instructions are also provided in the T10000 instruction manual.

1. Check to see if the following file is included in the development software that was installed.
/usr/local/sce/rpm/dsnet-X.X.XX-X.i386.rpm
("X.X.XX-X" indicates the version number)
2. Use a web browser to access the DTL-T10000.
3. Select "Packages" from the Settings menu.
4. Enter the host name or IP address of the development computer, and a user name and password for logging into the development computer. Enter "/usr/local/sce/rpm" in the "Directory" field.
5. When a list of packages is displayed, select the file shown above and click the "Install" button.
6. Select "Shutdown" from the Settings menu and reboot the DTL-T10000.

Environment Settings

Path settings

The following directories must be added to the path.

Add the following to the lines setting up the path in the environment settings file of the shell (.bash_profile or .cshrc).

```
/usr/local/sce/bin
/usr/local/sce/ee/gcc/bin
/usr/local/sce/iop/gcc/bin
```

Environment Variable Settings

By specifying the T10000 hostname (or IP address) in the DSNETM environment variable, the debugger (dsedb), etc. can be run without specifying the -d option.

Make the following additions to the shell environment settings file (e.g. .bash_profile, cshrc).

```
for .bash_profile(.bashrc): $ export DSNETM=hostname
for .cshrc:      setenv DSNETM hostname
```

Writing to Flash ROM

Of the DTL-T10000's internal software, the Flash ROM, which corresponds to the kernel of the PlayStation 2 is distributed in its newest release as an image file accompanying the development software. The Flash ROM can be updated using the following procedure.

1. Check to see that the following file is included in the development software that was installed.

```
/usr/local/t10000-XXXX.bin
("XXXX" indicates the version number)
```

2. Run dsflash.

```
$ dsflash /usr/local/t10000-XXXX.bin
*** Updating Flash ROM ...
*** Resetting...
```

MPU-4.0 board DSW602

3. When the program finishes and displays "Complete!", run dsreset.

```
$ dsreset
dsreset (Version X.X.XX...)
$
```

Confirming the Flash ROM Version

Starting dsibd will display the message shown below. Confirm the underlined portion.

```
CPUID=15, ROMGEN=2001-0219, CACH_CONFIG=1edd8, 8MB, IOP mode,
FlashROM boot <20010219-213228,ROMconf,t10000-relXX.bin:11824>
```

If dsflash is interrupted during processing or if the write fails for some reason, there is a chance that writing will not work when dsflash is re-executed. If this happens, download the DTL-T10000 kernel repair tool from the SCEA or SCEE developer support website and execute the repair process.

Compiling and Running Sample Programs

The procedure for building sample programs and running them on the DTL-T10000 is described below. If the sample programs run properly, the development environment is set up correctly.

Changing Access Permission Attributes

After installing, the sce/ee/sample and sce/iop/sample directories will be read-only. To build sample programs, change the access permission attributes in the following manner.

```
$ su root
Password:xxxxxxxx <= enter root password
# cd /usr/local/sce/ee
# chmod -R a+w sample      <= make sample directory and subdirectories world-writeable
# exit
$
```

Make the same changes for the iop directory.

basic3d: Running an EE Sample Program

The procedure for building and running a program that is run on the EE will be described, using sce/ee/sample/basic3d/vu1 as an example.

Building the Sample Program

Build using 'make' or 'make all'.

The sample programs use general purpose Makefiles so they can be used for user programs as well by rewriting the TOP= and OBJS= lines, etc.

```
$ cd /usr/local/sce/ee/sample/basic3d/vu1
$ make all
```

Running the Sample Program

If the DSNETM environment variable described above is set up properly, the sample program can be run with 'make run'. If the variable is not set, the program can be run by indicating the T-10000 hostname, as shown below.

```
$ dsedb -d hostname -r run main.elf
```

The elf file is sent to the T-10000 via the network. If the program runs properly, it will produce the following output.

```
$ dsedb -d hostname -r run main.elf
*** Resetting...
EE DECI2 Manager version...
CPUID=..., BoardID=..., ROMGEN=..., 128M

Loading program (address=...size=...)...
Loading XXX symbols...
Entry address =...
GP value    =...
```

Operate the controller connected to the DTL-T10000.

To stop the sample program, press Ctrl+C from the console of the development computer. An image will stay on the monitor but this is normal.

sqhard: Running an IOP Sample Program

The procedure for building and running a program that runs on the IOP will be described, using sce/iop/sample/sound/sqhard, a sample program that performs MIDI playback using the hardware synthesizer.

1. Building sqhard

To build the program, use the make command from the directory containing the sample program.

```
$ cd /usr/local/sce/iop/sample/sound/sqhard  
$ make
```

2. Running sqhard

Run the built program using 'make run'.

```
$ make run
```

Once the module specified in the makefile is loaded, a tune will be played. To stop while the tune is playing, press Ctrl+C. If the tune continues to play, run dsreset.

sqsoft: Running an EE-IOP Sample Program

The procedure for building and running a program that runs on the EE and IOP will be described, using sce/iop/sample/sound/sqsoft, a sample program performing MIDI playback using the software synthesizer. The program operates together with sce/ee/sample/sound/sqsoft, which runs on the EE side.

1. Building the IOP-side module

First, the program module to be run on the IOP is built.

```
$ cd /usr/local/sce/iop/sample/sound/sqsoft  
$ make
```

2. Building the EE-side program

Next, the program to be run on the EE is built.

```
$ cd /usr/local/sce/ee/sample/sound/sqsoft  
$ make
```

3. Checking sound element data

Check to see that sound element data files are present.

```
$ ls /usr/local/sce/data/sound
```

4. Running the sample program

Run the EE-side program that was built.

```
$ make run
```

main.elf is sent to the DTL-T10000 and executed, the IOP-side module is loaded, sound element data is loaded, and a tune is played. Press Ctrl+C from the console on the development computer to stop the tune. If the tune continues to play, run dsreset.

As described above, the method used to execute sample code depends on the individual sample. For details, read the readme.txt files in the sample directories.

